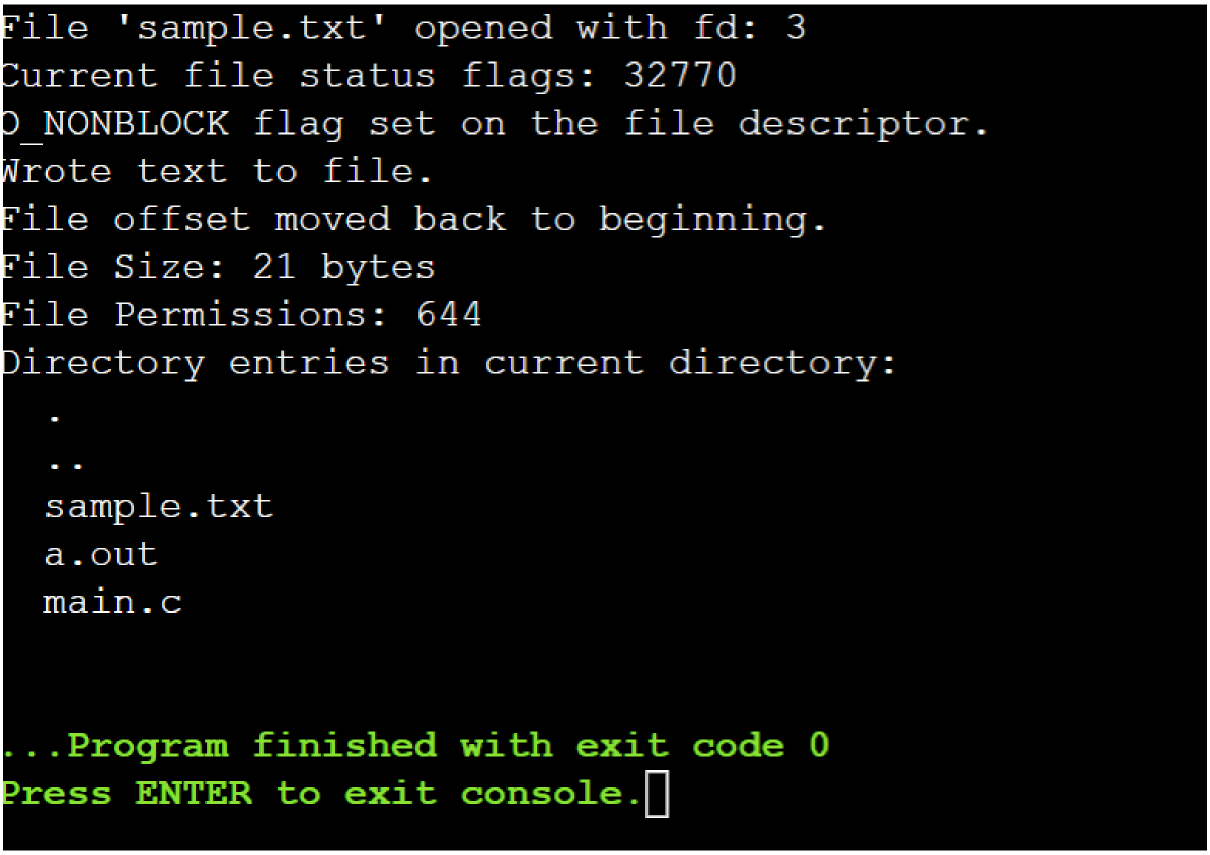
```
    }

#include <stdio.h>
#include <fcntl.h>      // fcntl()
#include <unistd.h>     // lseek(), close()
#include <sys/stat.h>   // stat()
#include <dirent.h>     // opendir(), readdir()
#include <stdlib.h>
#include <string.h>
#include <errno.h>

int main() {
    // 1. Open a file and use fcntl to get/set flags
    int fd = open("sample.txt", O_CREAT | O_RDWR, 0644);
    if (fd < 0) {
        perror("open");
        return 1;
    }

    printf("File 'sample.txt' opened with fd: %d\n", fd);

    // Get current flags
    int flags = fcntl(fd, F_GETFL);
    if (flags == -1) {
        perror("fcntl - F_GETFL");
        close(fd);
        return 1;
    }
    printf("Current file status flags: %d\n", flags);
```

```
// Set non-blocking flag (just as an example)

if (fcntl(fd, F_SETFL, flags | O_NONBLOCK) == -1) {

    perror("fcntl - F_SETFL");

    close(fd);

    return 1;

}

printf("O_NONBLOCK flag set on the file descriptor.\n");


// 2. Write something and use lseek to move file pointer

const char *text = "This is a test file.\n";

if (write(fd, text, strlen(text)) == -1) {

    perror("write");

    close(fd);

    return 1;

}

printf("Wrote text to file.\n");


// Move offset back to beginning

if (lseek(fd, 0, SEEK_SET) == -1) {

    perror("lseek");

    close(fd);

    return 1;

}

printf("File offset moved back to beginning.\n");


// 3. Use stat() to get file info

struct stat fileStat;

if (stat("sample.txt", &fileStat) == -1) {
        perror("stat");
        close(fd);
        return 1;
    }
    printf("File Size: %ld bytes\n", fileStat.st_size);
    printf("File Permissions: %o\n", fileStat.st_mode & 0777);

    // 4. Read directory contents using opendir() and readdir()
    DIR *dir = opendir(".");
    if (!dir) {
        perror("opendir");
        close(fd);
        return 1;
    }
    printf("Directory entries in current directory:\n");

    struct dirent *entry;
    while ((entry = readdir(dir)) != NULL) {
        printf("  %s\n", entry->d_name);
    }
    closedir(dir);

    // Close the file descriptor
    close(fd);

    return 0;
}
```



```
File 'sample.txt' opened with fd: 3
Current file status flags: 32770
O_NONBLOCK flag set on the file descriptor.
Wrote text to file.
File offset moved back to beginning.
File Size: 21 bytes
File Permissions: 644
Directory entries in current directory:
  .
  ..
  sample.txt
  a.out
  main.c


...Program finished with exit code 0
Press ENTER to exit console.
```